AD P000972

Copyright 1982 by Elsevier Science Publishing Company, Inc.
NUMERICAL GRID GENERATION
Joe F. Thompson, editor

TRANSFINITE MAPPINGS AND THEIR APPLICATION TO GRID GENERATION

WILLIAM J. GORDON AND LINDA C. THIEL
Department of Mathematical Sciences, Drexel University,
Philadelphia, PA 19104 USA

SUMMARY

The two essential ingredients of any boundary value problem are the field equations which describe the physics of the problem and a set of relations which specify the geometry of the problem domain. Mesh generators or grid generators are preprocessors which decompose the problem domain into a large number of interconnected finite elements or curvilinear finite difference stencils. A number of such techniques have been developed over the past decade to alleviate the frustration and reduce the time involved in the tedious manual subdividing of a complex-shaped region or 3-D structure into finite elements. Our purpose here is to describe how the techniques of bivariate and trivariate "blending function" interpolation, which were originally developed for and applied to geometric problems of computer-aided design of sculptured surfaces and 3-D solids, can be adapted and applied to the geometric problems of grid generation. In contrast to other techniques which require the numerical solution of complex partial differential equations (and, hence, a great deal of computing), the transfinite methods proposed herein are computationally inexpensive.

1. INTRODUCTION

Over the past decade, a number of schemes have been developed for automating the generation of finite element and curvilinear finite difference grids. Among these, the *transfinite mapping* technique of Gordon and Hall[5] has been shown to have a number of advantages (cf. [6],[7]). Some of these are:

1. Exact modeling of boundaries
2. Minimal input effort
3. Automatic node connectivity definition
4. Well-suited to interactive graphics implementation
5. Good correlation between boundary nodes and interior mesh
6. Computationally efficient
7. Easy extension to three dimensions.

We use the term "*transfinite*" to describe this class of techniques since, unlike classical methods of higher dimensional interpolation which match the *primitive function* $\vec{P}$ at a finite number of points, the transfinite methods match

$\vec{F}$ at a nondenumerable number of points. In fact, as we shall see below, transfinite mappings of the plane match $\vec{F}$ along entire curve segments, while transfinite mappings in Euclidean 3-space can match $\vec{F}$ exactly on the six faces of a curvilinear parallelpiped.

To begin, we recall the geometric interpretation of the graph of a vector-valued function of two independent variables s and t

$$\vec{F}(s,t) = [x_1(s,t), x_2(s,t), \ldots, x_n(s,t)]^T. \tag{1}$$

As the variables s and t range over a domain S in the s,t-plane R^2, $F(s,t)$ traces out a region R in Euclidean n-space, E^n. That is, $\vec{F}$ maps regions in R^2 into regions in E^n

$$\vec{F}: \; R^2 \to E^n. \tag{2}$$

For two-dimensional problems, we shall be concerned with continuous transformations $\vec{F}$ which map the unit square $S = [0,1]x[0,1]$ one-to-one onto a simply connected, bounded region R in E^2 or E^3. Such maps can be thought of as topological distortions of the planar region S onto the two-dimensional manifold R, which is either a planar region ($R \subset E^2$) or a surface embedded in 3-space ($R \subset E^3$). In either case, a one-to-one (univalent) mapping $S \to R$ is equivalent to the introduction of a *curvilinear co-ordinate system* on R. The curve of constant *generalized co-ordinate* $s = s^*$ is the image $\vec{F}(s^*,t)$ of the line $s = s^*$ in S.

Similarly, the curve $\vec{F}(s,t^*)$ is the set of all points in R with generalized co-ordinate $t = t^*$. Thus, the point $\vec{F}(s^*,t^*)$ on R is said to have generalized co-ordinates (s^*,t^*), and, since the mapping $S \to R$ is univalent, any point $P \in R$ can be uniquely referenced by its generalized co-ordinates.

If S is the unit cube $[0,1]x[0,1]x[0,1]$ in the s,t,u-parameter space R^3 and R is a bounded region in Euclidean 3-space, then a one-to-one mapping $\vec{F}$ of S onto R can be envisioned as a topological distortion of the cube into R. Such a mapping of $R^3 \to E^3$ generates a curvilinear co-ordinatization of the solid R so that each point of R may be referenced by its *generalized coordinates* (s,t,u).

For bounded, simply connected planar domains R, one could of course generate an orthogonal co-ordinatization by means of a conformal mapping of R onto a canonical region such as a square or a circle. *However, from a practical standpoint, the construction of a conformal map is equivalent to the solution of Laplace's equation and is thus contrary to the goal of computational simplicity.* In contrast, the transfinite mappings described below are relatively simple to construct and implement for a wide variety of regions, and are computationally inexpensive.

2. TWO-DIMENSIONAL REGIONS

We first consider the case in which S is the unit square $[0,1]\times[0,1]$. Let us postulate the *existence* of a *primitive function* F which maps S onto R. It should be remarked that the function $\vec{F}$ is a fiction which we introduce only for notational simplicity and convenience. In practice, the only thing we are given is the geometric description of R in terms of its boundary, and it is the task of the analyst to cast this information into a form appropriate to the mapping formulas considered below. This, however, is not difficult to do and can be implemented by a computer subroutine.

Generically, $\vec{F}$: $R^2 \to E^2$ should be thought of as a continuous vector-valued function of the two independent variables s and t such that $\vec{F}$: $\partial S \to \partial R$. For example, consider the following mapping:

$$\vec{F}(s,t) = \begin{pmatrix} x(s,t) \\ y(s,t) \end{pmatrix} = \begin{pmatrix} 4st(1-s)(1-t) + (1+t/\sqrt{2}) \cos\frac{s\pi}{2} \\ (1+t/\sqrt{2}) \sin\frac{s\pi}{2} \end{pmatrix} \tag{3}$$

This maps the unit square $[0,1]\times[0,1]$ onto the quarter annulus R shown in Fig.1. The perimeter of the unit square maps onto the perimeter of R, and lines of constant s and constant t map onto the curvilinear co-ordinate system illustrated. In other words, each of the curves shown in the figure is a curve of *generalized co-ordinate* s = const. or t = const.

Our problem is to construct a univalent (one-to-one) function $\vec{U}$: $S \to R$ which matches $\vec{F}$ on the boundary of S, i.e.

$$\left.\begin{aligned} \vec{U}(0,t) &= \vec{F}(0,t),\ \vec{U}(s,0) = \vec{F}(s,0) \\ \vec{U}(1,t) &= \vec{F}(1,t),\ \vec{U}(s,1) = \vec{F}(s,1) \end{aligned}\right\} \tag{4}$$

A function $\vec{U}$ which interpolates to $\vec{F}$ at a non-denumerable set of points as in (4) will be termed a *transfinite interpolant* of $\vec{F}$.

To explain the notion of transfinite mapping, we shall find it convenient to rely on the *algebraic theory* of approximation developed in [3] and [4]. In this paper, by a *projector* P we shall mean an idempotent linear operator whose domain is the linear space F of all continuous functions defined on S and whose range is a subspace of F. The above interpolation problem (4) can be viewed as a search for a projector P such that $\vec{U} = P[\vec{F}]$ is a univalent map of $S \to R$ which satisfies the desired interpolatory properties. $\vec{U}$ is termed the *projection* of $\vec{F}$ or the *image* of $\vec{F}$ *under* P.

Suppose now that ϕ_0, ϕ_1 and ψ_0, ψ_1 are four univariate functions which satisfy the *cardinality conditions*

$$\left.\begin{aligned} \phi_i(k) = \delta_{ik} \equiv \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases} \quad & \text{for } i,k = 0,1 \\ \psi_j(\ell) = \delta_{j\ell} \quad & \text{for } j,\ell = 0,1 \end{aligned}\right\} , \tag{5}$$

and consider the *projectors* P_s and P_t defined by

$$\left.\begin{aligned} P_s[\vec{F}] &\equiv \phi_0(s)\vec{F}(s_0,t) + \phi_1(s)\vec{F}(s_1,t) \\ P_t[\vec{F}] &\equiv \psi_0(t)\vec{F}(s,t_0) + \psi_1(t)\vec{F}(s,t_1) \end{aligned}\right\}^{*} . \tag{6}$$

Then, the *product* projection

$$P_sP_t[\vec{F}] = \sum_{i=0}^{1}\sum_{j=0}^{1} \phi_i(s)\psi_j(t)\vec{F}(s_i,t_j) \tag{7}$$

interpolates to $\vec{F}$ at the *four corners* of [0,1]x[0,1] and the *Boolean sum* projection

$$(P_s \oplus P_t)[\vec{F}] \equiv P_s[\vec{F}] + P_t[\vec{F}] - P_sP_t[\vec{F}] \tag{8}$$

interpolates to $\vec{F}$ on the *entire boundary* of [0,1]x[0,1]. These properties of the functions (7) and (8) may be readily verified by evaluating the right-hand sides for the appropriate values of s and t and recalling the cardinality properties (5); see also [3] or [4].

The functions ϕ_i and ψ_j in the above formulae are as yet unspecified except for their values at the points $s_0 = t_0 = 0$ and $s_1 = t_1 = 1$. They are commonly referred to as *'blending functions'* and the function $\vec{U} = (P_s \oplus P_t)[\vec{F}]$ is termed a blended interpolant. The simplest choice for the blending functions in (5) is the set of four linear functions

$$\left.\begin{aligned} \phi_0(s) &= 1 - s, & \psi_0(t) &= 1 - t \\ \phi_1(s) &= s, & \psi_1(t) &= t \end{aligned}\right\} . \tag{9}$$

The vector-valued bivariate function $\vec{U} \equiv (P_s \oplus P_t)[\vec{F}]$ obtained by using (8) and (9) is termed the *bilinearly blended interpolant* of $\vec{F}$, or the *transfinite bilinear interpolant* of $\vec{F}$. Explicitly, it is given by

$$\begin{aligned} \vec{U}(s,t) = (1-s)\vec{F}(0,t) &+ s\vec{F}(1,t) + (1-t)\vec{F}(s,0) + t\vec{F}(s,1) \\ &- (1-s)(1-t)\vec{F}(0,0) - (1-s)t\vec{F}(0,1) \\ &- s(1-t)\vec{F}(1,0) - st\vec{F}(1,1) . \end{aligned} \tag{10}$$

This function has the properties that $\vec{U} = \vec{F}$ on the perimeter of the unit square [0,1]x[0,1]. This was first demonstrated by S.A. Coons in [2].

Figure 2 illustrates the mappings induced by the projectors (6)-(8) on a

*The reader should verify that both the operators P_s and P_t are, in fact, projectors, i.e., they are linear and idempotent.

region R with blending functions given by (9). It is readily seen that $P_s[\vec{F}]$ and $P_t[\vec{F}]$ each match only two opposing boundary segments, $P_sP_t[\vec{F}]$ matches only the corners of R, but $(P_s \oplus P_t)[\vec{F}]$ does, in fact, interpolate the complete perimeter of R. The mappings $P_s[\vec{F}]$ and $P_t[\vec{F}]$ are sometimes referred to as "linear lofting", in analogy with the drafting procedure known as lofting. $P_sP_t[\vec{F}]$, which is linearly ruled in both directions, is termed bilinear, and $(P_s \oplus P_t)[\vec{F}]$ is properly termed *bilinearly blended*. The three projectors P_s, P_t and $P_s \oplus P_t$ are all *transfinite projectors* since they interpolate $\vec{F}$ at a nondenumerable number of points.

Other examples of transfinite mappings obtained using equation (10) are shown in Figures 3,4 and 5.

We can generalize the above notions in two ways: first, we may consider mappings of $R^2 \to E^n$ for general n; and secondly, we may interpolate $\vec{F}$ *not only* on the boundary of the region $R = \{(x_1,x_2,\ldots,x_n)^T = \vec{F}(s,t): 0 \le s, t \le 1\}$, but also along other 'flow lines' or constant generalized co-ordinate lines. To this end, let $0 < s_0 < s_1 < \ldots < s_M = 1$ and $0 = t_0 < t_1 < \ldots < t_N = 1$, and let $\{\phi_i(s)\}_{i=0}^{M}$ and $\{\psi_j(t)\}_{j=0}^{N}$ be functions satisfying

$$\phi_i(s_k) = \delta_{ik}, \quad \psi_j(t_\ell) = \delta_{j\ell} \tag{11}$$

$0 \le i, k \le M$, $0 \le j, \ell \le N$. Now define the projections

$$\left.\begin{aligned} P_s[\vec{F}] &\equiv \sum_{i=0}^{M} \phi_i(s)\vec{F}(s_i,t) \\ P_t[\vec{F}] &\equiv \sum_{j=0}^{N} \psi_j(t)\vec{F}(s,t_j) \end{aligned}\right\} \tag{12}$$

The *product projection*

$$P_sP_t[\vec{F}] \equiv \sum_{i=0}^{M}\sum_{j=0}^{N} \phi_i(s)\psi_j(t)\vec{F}(s_i,t_j) \tag{13}$$

interpolates to $\vec{F}$ on the finite point set $\{(s_i,t_j)\}_{i=0,j=0}^{M,N}$ while the *Boolean sum* or *transfinite interpolant*

$$(P_s \oplus P_t)[\vec{F}] \equiv P_s[\vec{F}] + P_t[\vec{F}] - P_sP_t[\vec{F}] \tag{14}$$

interpolates to $\vec{F}$ along the M+N+2 lines $s = s_i$, $0 \le i \le M$ and $t = t_j$, $0 \le j \le N$. That is, if $\vec{U}(s,t) \equiv (P_s \oplus P_t)[\vec{F}]$ then

$$\begin{aligned} \vec{U}(s,t_j) &= \vec{F}(s,t_j), \quad 0 \le j \le N \\ \vec{U}(s_i,t) &= \vec{F}(s_i,t), \quad 0 \le i \le M \end{aligned} \tag{15}$$

If $M = N = 1$ and $s_0 = t_0 = 0$, $s_1 = t_1 = 1$, (14) reduces to the *transfinite bilinear interpolant* (10). If $M = N = 2$ and $s_0 = t_0 = 0$, $s_1 = t_1 = 1/2$, $s_2 = t_2 = 1$, then using the blending functions

$$\phi_0(s) = 2(s-1/2)(s-1), \quad \psi_0(t) = 2(t-1/2)(t-1)$$
$$\phi_1(s) = 4s(1-s), \quad \psi_1(t) = 4t(1-t) \tag{16}$$
$$\phi_2(s) = 2s(s-1/2), \quad \psi_2(t) = 2t(t-1/2)$$

in (14) yields the *biquadratically blended interpolant* of $\vec{F}$ along the *six lines* $s = 0, 1/2, 1$ and $t = 0, 1/2, 1$.

There are cases in which bilinear transfinite mapping techniques will not produce a satisfactory curvilinear grid. This typically happens on regions R which are so grossly distorted that there is either too great a variation in the size of grid elements or protions of the generalized co-ordinate curves actually map outside the region ("overspill", cf. [5]). An example is shown in Figure 6. Here, a bilinear transfinite mapping was executed with the result that some constant generalized co-ordinate lines overspilled the region.

There are basically three ways to deal with such difficulties. First, one may decompose the overly complex region into two or more simpler subregions and map each of these separately. Although this approach generally works well, there may be problems at the interfaces between subregions, since the generalized co-ordinate lines will have slope discontinuities there. One way of handling this difficulty is to employ higher degree blending functions, e.g., cubic Hermite blending functions.

A second way of attempting to achieve a satisfactory transformation is to reparametrize the boundary segments of R by, for example, introducing monotonic transformations of the independent variables s and/or t.

Another way of dealing with complex regions is to introduce *auxiliary constraints* into the transformation problem. Since the paramount objective is to obtain a one-to-one (invertible) mapping of S onto R, the analyst is perfectly free to enforce whatever additional constraints he feels will guarantee the invertibility of the mapping. In our experience, we have generally found it adequate to specify, as an auxiliary constraint, the image (i.e., the mapped position) of a single interior point of S. That is, we identify where inside R we desire to map a selected point in the interior of S. For simplicity, suppose the point in S whose image position we want to constrain is the mid-point of the square, $s = t = 1/2$. We want to force this point to map into the point in R having co-ordinates (α,β). Let $\vec{U}$ be the bilinearly blended function of (10). Then, the following transformation maps ∂S onto ∂R and maps $(1/2,1/2)$ onto the point (α,β):

$$\vec{V}(s,t) = \vec{U}(s,t) + 16s(1-s)t(1-t)\left[\begin{pmatrix}\alpha\\ \beta\end{pmatrix} - \vec{U}(1/2,1/2)\right]. \tag{17}$$

To verify this, note that along the perimeter of $[0,1]\times[0,1]$ the right-hand side

reduces to just $\vec{U}(s,t)$, which maps ∂S exactly onto ∂R, as desired. For $(s,t) = (1/2,1/2)$, the right-hand side reduces to just (α,β). More generally, the point $(s,t) = (a,b)$ can be mapped into (α,β) by the formula:

$$\vec{V}(c,t) = \vec{U}(s,t) + \frac{s(1-s)t(1-t)}{a(1-a)b(1-b)}\left[\begin{pmatrix}\alpha\\ \beta\end{pmatrix} - \vec{U}(s,t)\right] . \tag{18}$$

As an example, Figure 6 shows a region R for which the bilinearly blended transformation (10) does not give an invertible mapping of S onto R. It is intuitively clear that the image of the point $(s,t) = (1/2,1/2)$ has mapped too far to the right. Therefore, we enforce the auxiliary constraint that the point $(1/2,1/2)$ in S should map onto the point $(.494,.119)$ in R. Figure 7 illustrates the result of the transformation obtained using (17).

In [5], Gordon and Hall propose the use of more general auxiliary constraints. For example, instead of just a single point, they consider mappings for which lines of constant s or t are forced to map onto preselected generalized co-ordinate curves in the domain R. Such curves may arise naturally as interfaces between subregions of R, or they may be determined by the analyst on geometric grounds. The transformation formulas appropriate to these constrained maps are given by equation (14).

If the region R is basically triangular, in contrast to quadrilateral transfinite interpolation techniques over triangles may be more appropriate. The theory for such "trilinearly blended" methods was developed in [1]. The details of these techniques as applied to grid generation may be found in [6] and [7].

As a practical matter, the curves bounding R may not be easily represented as closed-form mathematical expressions. Nevertheless, the above results still apply if the boundary curves are represented as discrete point sets, i.e., piecewise linear curves. For a fuller discussion of "*discretized transfinite mappings*" see [6] and [7]. Surfaces in Euclidean 3-space are handled in precisely the same way as 2-D regions. All that need be done is to add the third co-ordinate functions $Z(s)$ and $Z(t)$ to the x and y components. A discussion of surface decomposition techniques is given in [7].

3. THREE-DIMENSIONAL SOLID STRUCTURES

The purpose of this section is to outline the extensions to 3-dimensions of the bivariate transfinite mapping techniques discussed above. To begin, we consider the following three projectors:

$$\begin{aligned} P_s[\vec{F}] &= (1-s)\vec{F}(0,t,u) + s\vec{F}(1,t,u) \\ P_t[\vec{F}] &= (1-t)\vec{F}(s,0,u) + t\vec{F}(s,1,u) \\ P_u[\vec{F}] &= (1-u)\vec{F}(s,t,0) + u\vec{F}(s,t,1) \ . \end{aligned} \tag{19}$$

In these expressions, the "primitive function" $\vec{P}$ is a vector-valued function of the three independent parameters s,t and u. As s,t and u range over the unit cube, $\vec{F}$ maps out the solid volume R. Depending upon the co-ordinate system used, the three components of $\vec{F}$ may correspond to Cartesian, spherical, cylindrical, toroidal, etc. co-ordinates. As a practical matter, the co-ordinate system employed should be that which most appropriately fits the geometry and topology of the problem domain.

Quite clearly, the three projectors in (19) correspond, respectively, to linear blending (lofting) in s,t and u. Now, however, the geometric entities on the right-hand side of the expressions are not curves, as in (6), but rather *surfaces*. For example, as t and u range over the parameter domain [0,1]x[0,1], the vector-valued function F(0,t,u) traces out a surface in Euclidean 3-space corresponding to one of the six faces of the solid volume R.

It is useful to consider the pairwise products of the above three projectors. For instance, the product of the first and the second is

$$\begin{aligned} P_s(P_t[\vec{F}]) = {} & (1-s)(1-t)\vec{F}(0,0,u) + (1-s)t\vec{F}(0,1,u) \\ & + s(1-t)\vec{F}(1,0,u) + st\vec{F}(1,1,u) \end{aligned} \tag{20}$$

The right-hand side of this expression contains four expressions which refer to the *edges* of the object under consideration. By evaluating the expression along the four edges (s,t) = (0,0),(0,1),(1,0) and (1,1), it can be verified that the trivariate function $P_sP_t[\vec{F}]$ does, in fact, match $\vec{F}$ along these edges. (It may be easily demonstrated that $P_sP_t[\vec{F}] = P_tP_s[\vec{F}]$, i.e., the projectors commute, just as in the bivariate case.)

We have seen that the projectors P_s,P_t and P_u each interpolate the two opposing *faces* of the solid volume described by $\vec{F}(s,t,u)$, and that products of pairs of these projectors interpolate to the *edges* of the solid. If we take the product of all three of the projectors in (19), we obtain the expression:

$$\begin{aligned} P_sP_tP_u[\vec{F}] = {} & (1-s)(1-t)(1-u)\vec{F}(0,0,0) + (1-s)(1-t)u\vec{F}(0,0,1) \\ & + (1-s)t(1-u)\vec{F}(0,1,0) + (1-s)tu\vec{F}(0,1,1) \\ & + s(1-t)(1-u)\vec{F}(1,0,0) + s(1-t)u\vec{F}(1,0,1) \\ & + st(1-u)\vec{F}(1,1,0) + stu\vec{F}(1,1,1). \end{aligned} \tag{21}$$

The right-hand side of this expression contains values of $\vec{F}$ which refer to the eight *corners* of the region R. It does, in fact, interpolate to these eight corners. Since (21) is linear in each of the three parameters s,t and u, it is termed a *trilinear interpolant*. The graph of the trilinear interpolant is a six-sided polyhedron which passes through the vertices of R; it is simply the 3-D

generalization of a quadrilateral in Euclidean 2-space.

In the 2-dimensional case, one starts with the two projectors P_s and P_t of (6) and, by combination, generates a total of four; namely, P_s, P_t, P_sP_t and $P_s \oplus P_t$. In three dimensions, the situation is much more complex and the variety of possible projectors much richer. In [3] and [4], it is shown that under the two binarary operations of operator multiplication and Boolean ($\oplus$) addition, any collection of commutative projectors forms a *distributive lattice*. Space does not permit going into the details of this theory, but we can illustrate some of the results in the trivariate case. Without proof, we state that there are 21 distinct projectors which can be formed by multiplication and $\oplus$ addition of the three projectors P_s, P_t and P_u. In addition to those displayed above, the following are *examples:*

$$
\begin{aligned}
&P_s \oplus P_t = P_s + P_t - P_sP_t \\
&P_s \oplus P_tP_u = P_s + P_tP_u - P_sP_tP_u \\
&P_sP_t \oplus P_u = P_sP_t + P_u - P_sP_tP_u \\
&P_s \oplus P_t \oplus P_tP_u = P_s + P_t - P_sP_t \\
&P_sP_t \oplus P_tP_u \oplus P_uP_s = P_sP_t + P_tP_u + P_uP_s - 2P_sP_tP_u \\
&P_s \oplus P_t \oplus P_u = P_s + P_t + P_u - P_sP_t - P_tP_u - P_uP_s + P_sP_tP_u
\end{aligned}
\qquad (22)
$$

One should note that, because of the idempotency and linearity of the projectors, much cancellation occurs. For instance, one has $P_s \oplus P_sP_t = P_s + P_sP_t - P_sP_sP_t = P_s$, which means that nothing is achieved by taking the $\oplus$ sum of P_s and P_sP_t. This is because the interpolation properties of P_sP_t are a subset of those of the projector P_s.

An aspect of the theory developed in [3] and [4] is that there is an *isomorphism* between the distributive lattice of projectors and the associated distributive lattice of their precision sets. In other words, if we know the expression for a certain projector, then we can determine the point set on which it interpolates by replacing operator multiplication by set intersection and $\oplus$ addition by set union. For example, the *precision set* (set of points on which it interpolates) of P_s consists of the two faces of R defined by $s = 0$ and $s = 1$; and similarly for P_t and P_u. Thus, it follows from the isomorphism that the precision sets of the projectors in (22) are, respectively, given by the following expressions in which S_s, S_t and S_u are the precision sets of P_s, P_t and P_u:

$$
\begin{aligned}
&S_s \cup S_t \\
&S_s \cup (S_t \cap S_u) \\
&(S_s \cap S_t) \cup S_u \\
&S_s \cup S_t \cup (S_t \cap S_u) \\
&(S_s \cap S_t) \cup (S_t \cap S_u) \cup (S_u \cap S_s) \\
&S_s \cup S_t \cup S_u
\end{aligned}
\tag{23}
$$

If one thinks of the precision sets, it is obvious that the weakest ("*algebraically minimal*") of all projectors is the triple product $P_s P_t P_u$ and the "*algebraically maximal*" projector is the Boolean sum of all three: $P_s \oplus P_t \oplus P_u$. All other possible projectors are algebraically in-between these two.

Which of this miriad of 3-D projectors one uses in practice is a matter of what data is given; or, more precisely, *where* the data is given. In other words, one is given the precision set and must use the isomorphism "backward" to infer the appropriate projector (i.e., interpolation formula). For instance, if one does, in fact, know the exact shapes of all six of the bounding surfaces of R, then the appropriate mapping equation is the full-blown expression $(P_s \oplus P_t \oplus P_u)[\vec{F}]$, which explicitly involves all faces, all edges and all corners. At the other extreme, one may only know the co-ordinates of the eight corners of R. In this case, one would use the transformation equation (21).

In practice, the situation is usually somewhere in-between, i.e., the data is seldom as complete as required by the maximal projector $P_s \oplus P_t \oplus P_u$ or as scant as only the eight vertices needed in $P_s P_t P_u[\vec{F}]$. The examples given below assume that the function $\vec{F}$ is known (i.e., data is given) on the 12 edges of R. In this case, the revelant transformation equation or mapping formula is:

$$
\begin{aligned}
&(P_s P_t \oplus P_t P_u \oplus P_u P_s)[\vec{F}] \\
&\quad = (1-s)(1-t)\vec{F}(0,0,u) + (1-s)t\vec{F}(0,1,u) \\
&\quad + s(1-t)\vec{F}(1,0,u) + st\vec{F}(1,1,u) \\
&\quad + (1-t)(1-u)\vec{F}(s,0,0) + (1-t)u\vec{F}(s,0,1) \\
&\quad + t(1-u)\vec{F}(s,1,0) + tu\vec{F}(s,1,1) \\
&\quad + (1-s)(1-u)\vec{F}(0,t,0) + (1-s)u\vec{F}(0,t,1) \\
&\quad + s(1-u)\vec{F}(1,t,0) + su\vec{F}(1,t,1) \\
&\quad - 2[(1-s)(1-t)(1-u)\vec{F}(0,0,0) + (1-s)(1-t)u\vec{F}(0,0,1) \\
&\quad + (1-s)t(1-u)\vec{F}(0,1,0) + (1-s)tu\vec{F}(0,1,1) \\
&\quad + s(1-t)(1-u)\vec{F}(1,0,0) + s(1-t)u\vec{F}(1,0,1) \\
&\quad + st(1-u)\vec{F}(1,1,0) + stu\vec{F}(1,1,1)].
\end{aligned}
\tag{24}
$$

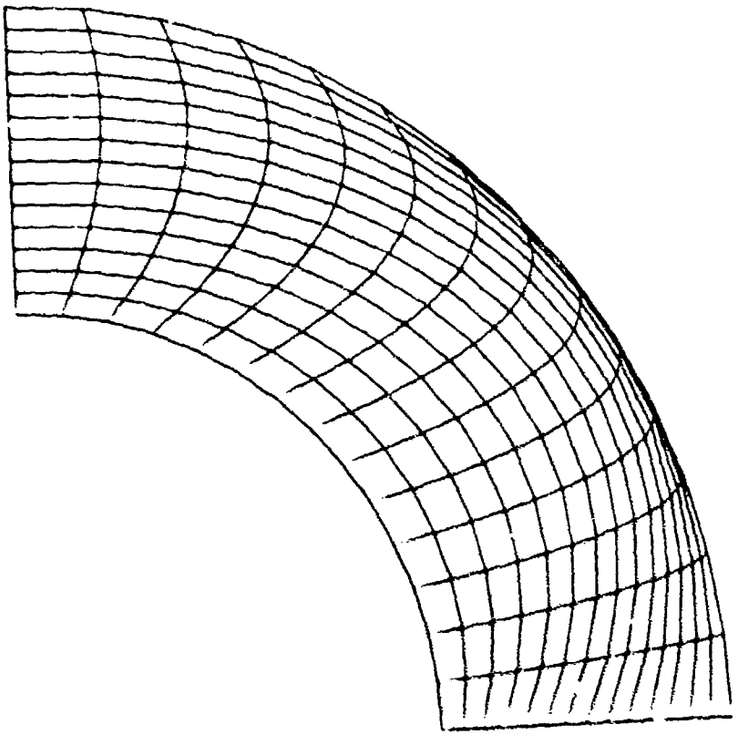
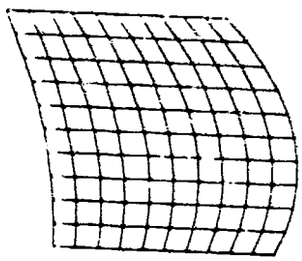


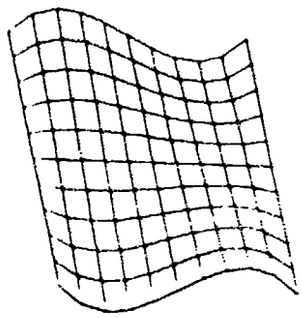
Figure 1

The curvilinear co-ordinate system induced by the following mapping:

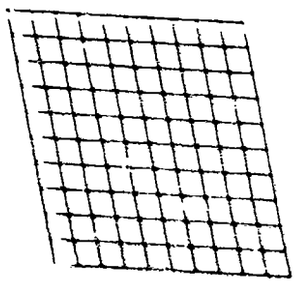
$$\vec{F}(s,t) = \begin{pmatrix} x(s,t) \\ y(s,t) \end{pmatrix} = \begin{pmatrix} 4st(1-s)(1-t)+(1+t\sqrt{2})\cos\frac{s\pi}{2} \\ (1+t\sqrt{2})\sin\frac{s\pi}{2} \end{pmatrix}$$



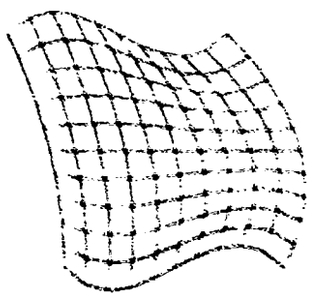
$P_S[\vec{F}]$



$P_t[\vec{F}]$



$P_S P_t[\vec{F}]$



$(P_S \bullet P_t)[\vec{F}]$

Figure 2

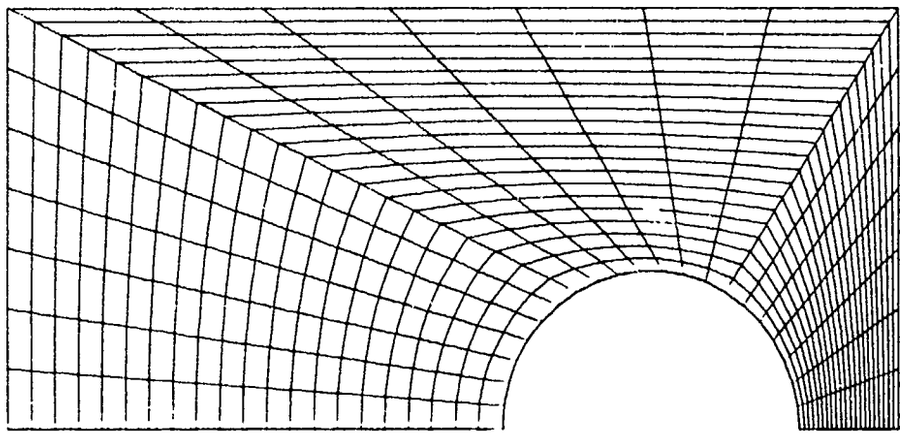


Figure 3

The curvilinear co-ordinate system generated by the following parametrization:

$$\vec{F}(0,t) = \begin{bmatrix} -2.5t + .5 \\ -2 \end{bmatrix}, \quad \vec{F}(1,t) = \begin{bmatrix} 0.5t + 2 \\ -2 \end{bmatrix}$$

$$\vec{F}(s,0) = \begin{bmatrix} 1.25 - .75 \cos(\pi s) \\ -2 + .75 \sin(\pi s) \end{bmatrix}$$

$$\vec{F}(s,1) = \begin{bmatrix} \begin{cases} -2, & 0.0 \le s < .33 \\ -6.5 + 13.5s, & .33 \le s < .66 \\ 2.5, & .66 \le s \le 1 \end{cases} \\ \begin{cases} -2 + 6s, & 0.0 \le s < .33 \\ 0.0, & .33 \le s < .66 \\ 4 - 6s, & .66 \le s \le 1 \end{cases} \end{bmatrix}.$$

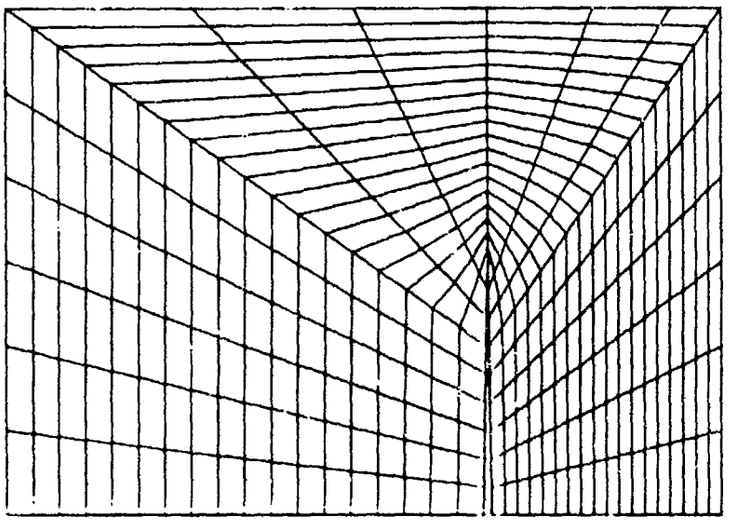


Figure 4

The transfinite map of a region with a crack via the following parametrization:

$$\vec{F}(0,t) = \begin{bmatrix} 1 - 3t \\ -3 \end{bmatrix}, \quad \vec{F}(1,t) = \begin{bmatrix} 1.05 + 1.45t \\ -3 \end{bmatrix}$$

$$\vec{F}(s,0) = \begin{bmatrix} 1 + .05s \\ \begin{cases} -3 + 3s, & 0.0 \le s \le 0.5 \\ -3s, & 0.5 \le s \le 1 \end{cases} \end{bmatrix},$$

$$\vec{F}(s,1) = \begin{bmatrix} \begin{cases} -2, & 0.0 \le s < .33 \\ -12.7 + 41.4s - 27.9s^2, & .33 \le s < .66 \\ 2.5, & .66 \le s \le 1 \end{cases} \\ \begin{cases} -3 + 9s & 0.0 \le s < .33 \\ 0.0 & .33 \le s < .66 \\ 6 - 9s & .66 \le s \le 1 \end{cases} \end{bmatrix}.$$

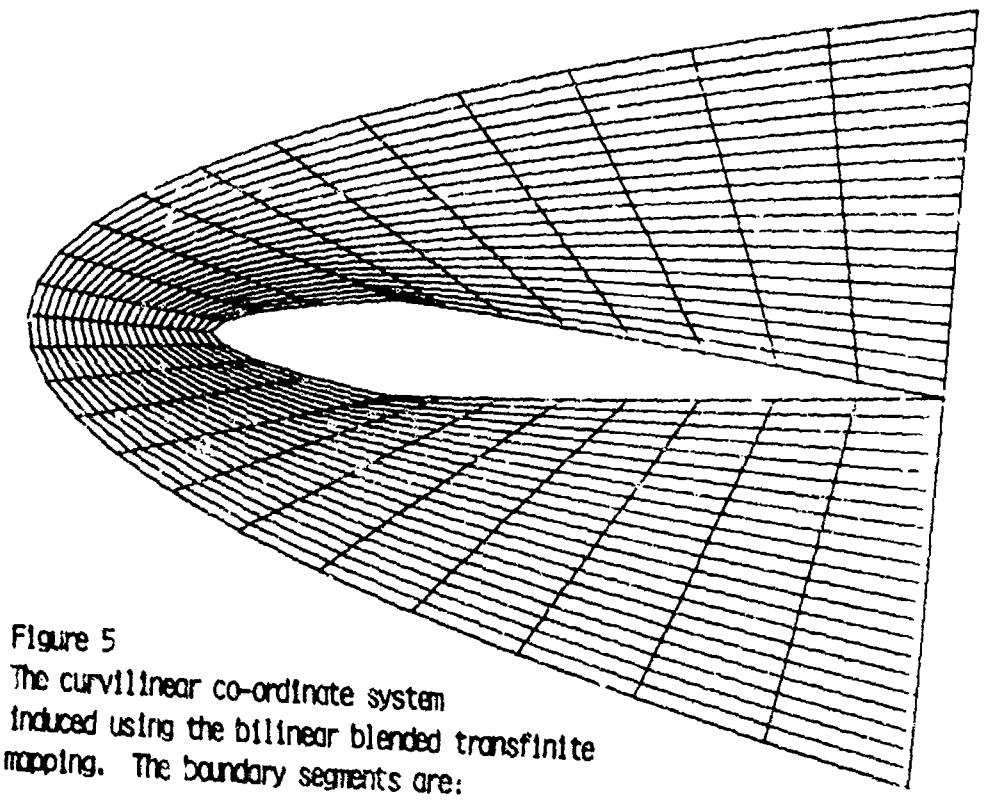


Figure 5
The curvilinear co-ordinate system induced using the bilinear blended transfinite mapping. The boundary segments are:

$$\vec{F}(0,t) = \begin{bmatrix} 4 \\ -4t \end{bmatrix}, \quad \vec{F}(1,t) = \begin{bmatrix} 4 \\ 4t \end{bmatrix},$$

$$\vec{F}(s,0) = \begin{bmatrix} \begin{cases} 4 - 11.5s + 7.5s^2, & 0.0 \le s < .33 \\ 9 - 36s + 36s^2, & .33 \le s < .66 \\ -3.5s + 7.5s^2, & .66 \le s \le 1 \end{cases} \\ \begin{cases} -1.125s + 1.125s^2, & 0.0 \le s < .33 \\ -.75 + 1.5s & .33 \le s < .66 \\ 1.125s - 1.125s^2 & .66 \le s \le 1 \end{cases} \end{bmatrix},$$

$$\vec{F}(s,1) = \begin{bmatrix} 4 - 20s + 20s^2 \\ -2 + 4s \end{bmatrix}.$$

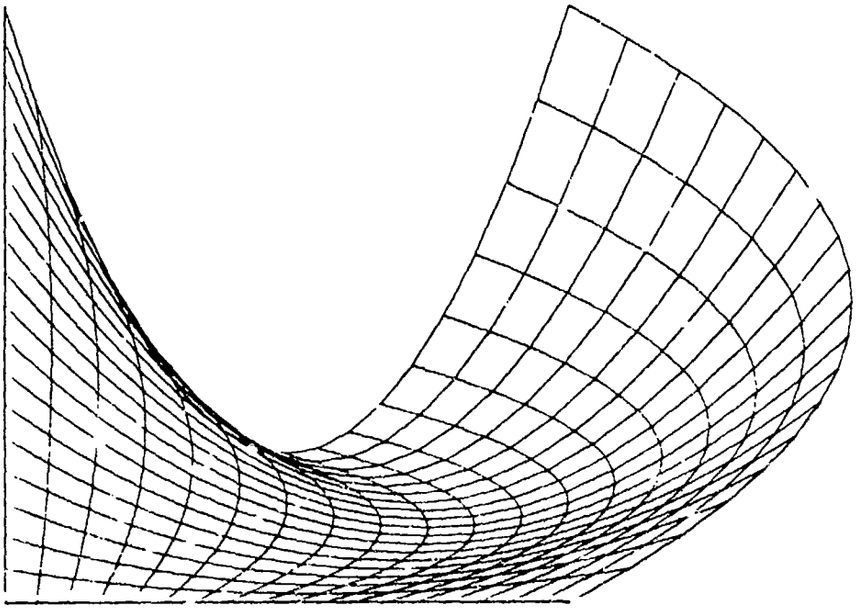


Figure 6
Bilinear blending here yields a non-univalent map. The parametrization of the boundary is:

$$\vec{F}(0,t) = \begin{bmatrix} 0 \\ t \end{bmatrix} \quad \vec{F}(1,t) = \begin{bmatrix} 1 + 2t - 2t^2 \\ t \end{bmatrix},$$

$$\vec{F}(s,0) = \begin{bmatrix} s \\ 0 \end{bmatrix}, \quad \vec{F}(s,1) = \begin{bmatrix} s \\ 1 - 3s + 3s^2 \end{bmatrix}.$$

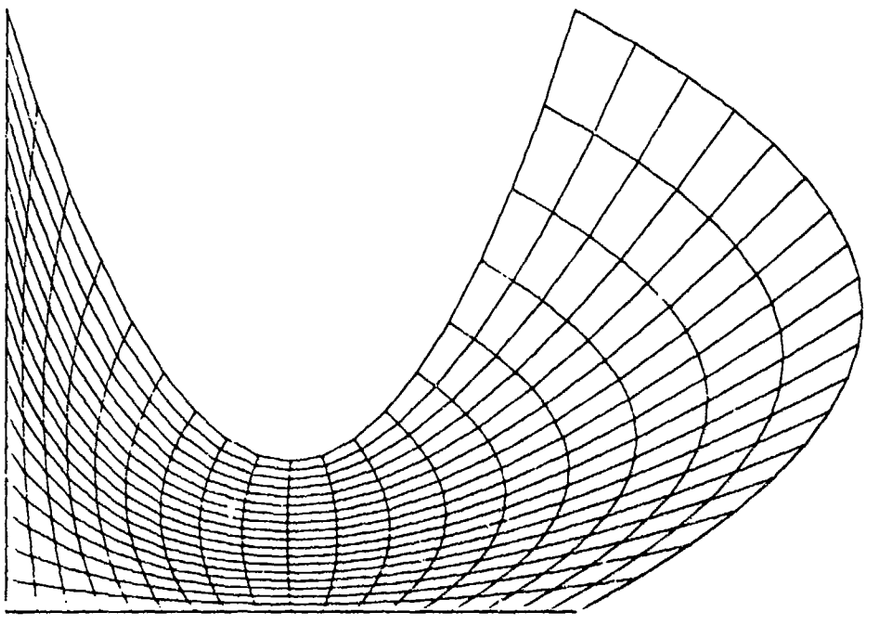


Figure 7
A curvilinear co-ordinate system with no "overspill" is achieved via (17). Here, the point (1/2,1/2) in the s,t-plane is mapped onto the point (.494, .119) in the x,y-plane.

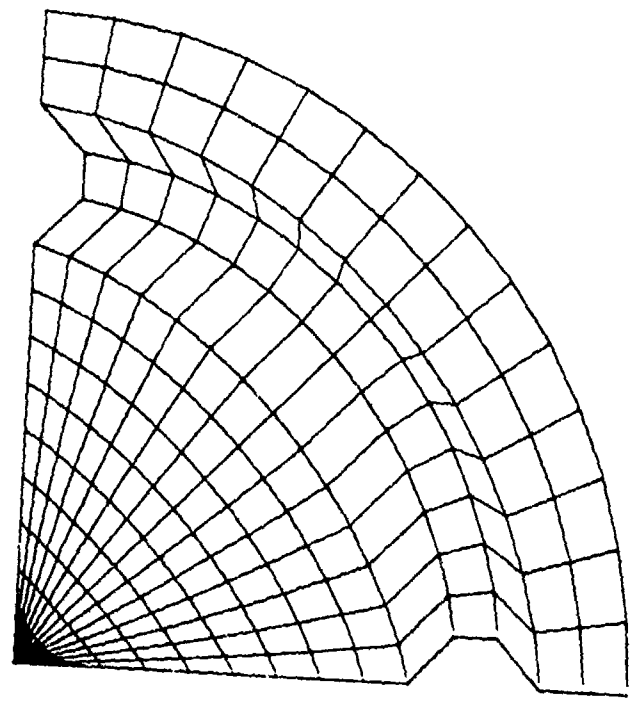


Figure 8

This mapping was generated via (18) by mapping the point (11/14,1/2) in the s,t-plane onto the point (.326,.334) in the x,y-plane. The boundary segments are:

$$\vec{F}(0,t) = \begin{bmatrix} -3 \\ -3 \end{bmatrix}, \quad F(1,t) = \begin{bmatrix} -3+6\cos(\pi t/2) \\ -3+6\sin(\pi t/2) \end{bmatrix}$$

$$\vec{F}(s,0) = \begin{bmatrix} 6s-3 \\ \begin{cases} -3+\sqrt{.25-(6s-4.5)^2}, & .67 \le s \le .83 \\ -3, & \text{otherwise} \end{cases} \end{bmatrix}$$

$$\vec{F}(s,1) = \begin{bmatrix} \begin{cases} -3+\sqrt{.25-(6s-4.5)^2}, & .67 \le s \le .83 \\ -3, & \text{otherwise} \end{cases} \\ 6s-3 \end{bmatrix}$$

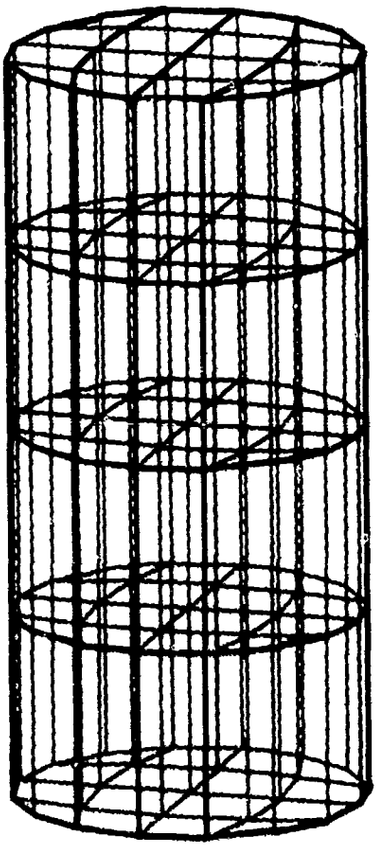


Figure 9

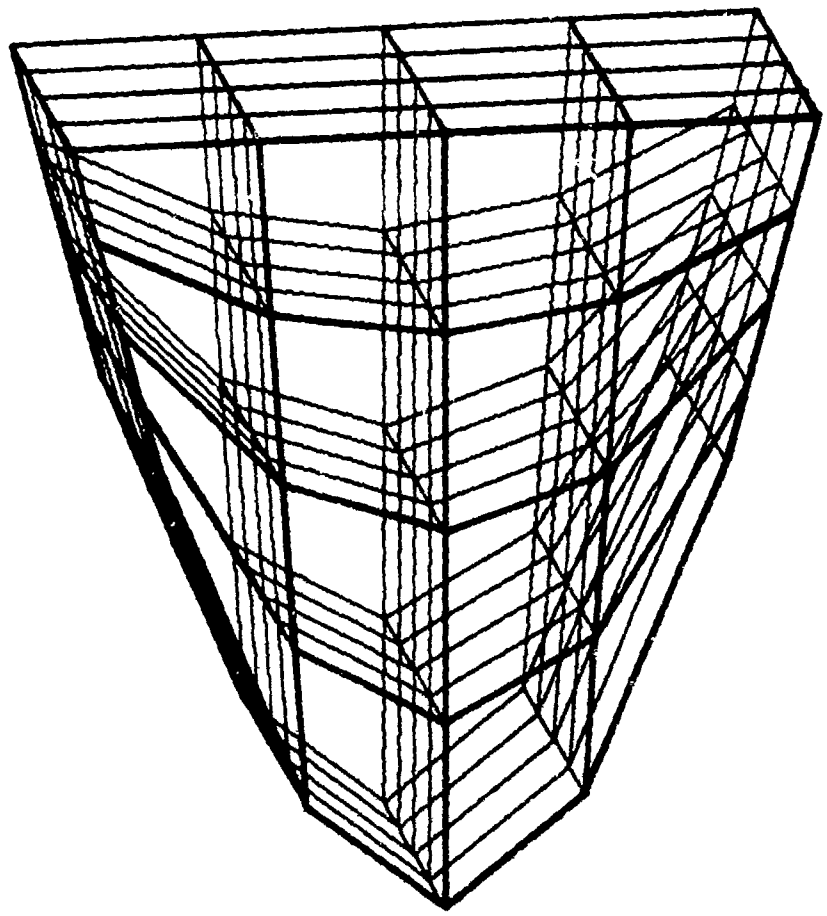


Figure 10

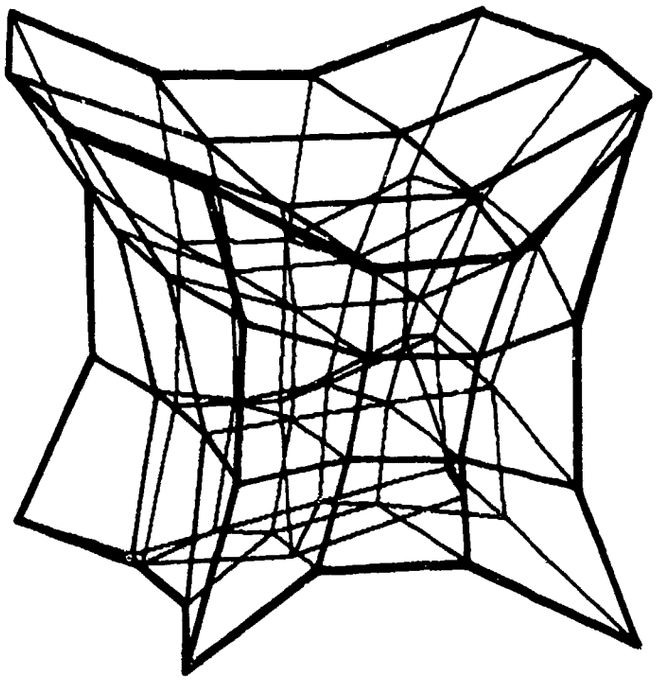


Figure 11

ACKNOWLEDGEMENTS

This work was supported in part by the U.S. Office of Naval Research and the Air Force Office of Scientific Research under contract #N00014-80-C0176 to Drexel University.

The authors wish to thank Debra DeLise-Hughes for her assistance in the preparation and typing of this paper, and Andrew Galardi for his help with some of the figures.

REFERENCES

1. G. Birkhoff, W.J. Gordon, R.E. Barnhill, "Smooth Interpolation in Triangles", J. Approx. Theory, 8, 114-128 (1973).
2. S.A. Coons, "Surfaces for Computer-Aided Design of Space Forms", Project MAC, Design Div., Dept. Mech. Engng., Mass. Inst. Tech. (1964). Available from: Clearinghouse for Federal Scientific-Technical Information, National Bureau of Standards, Springfield, VA, U.S.A.
3. W.J. Gordon, "Blending-Function Methods for Bivariate and Multivariate Interpolation and Approximation", SIAM J. Numer. Anal. 8, 158-177 (1971).
4. W.J. Gordon, "Distributive Lattices and the Approximation of Multivariate Functions", in Approximations with Special Emphasis on Spline Functions, Academic Press, New York, 1969, pp. 223-277.
5. W.J. Gordon, C.A. Hall, "Construction of Curvilinear Coordinate Systems and Applications to Mesh Generation", Inst. J. Num. Meth. Engng., 7, 461-477 (1973).
6. R.B. Haber, M.S. Shepard, J.F. Abel, R.H. Gallagher, and D.P. Greenberg, "A General Two-Dimensional Graphical Finite Element Preprocessor Utilizing Discrete Transfinite Mappings", Int. J. Num. Meth. Engng., 17 (7), 1015-1044 (1981).
7. R. Haber, J.F. Abel, "Discrete Transfinite Mappings for the Description and Meshing of Three-Dimensional Surfaces Using Interactive Computer Graphics", Int. J. Num. Meth. Engng., 18, 41-66 (1982).